

Letterboxd Requirements Specification

Author:	Prarthana Desai
Creation Date:	Nov 22, 2024
Last Updated:	Nov 22, 2024
Document Ref:	NA
Version:	1.2

Document Control

Change Record

Name	Position

Reviewers

Name	Position

Distribution

Name	Position

Note To Holders:

If you receive an electronic copy of this document and print it out, please write your name on the equivalent of the cover page, for document control purposes.

If you receive a hard copy of this document, please write your name on the front cover, for document control purposes.

Introduction

Letterboxd is a social networking platform for film enthusiasts to share and discover movies. This document outlines the essential features, user needs, and design guidelines for the project.

Functional Requirements:

1. User Registration and Login:

- Users can create an account using their email address
- Users can log in to their account using their credentials.
- The system will validate user input data (e.g., email address, password) to ensure correctness and security.

2. Movie Database:

- The platform will have a comprehensive database of movies, including:
 - Titles
 - Genres
 - Release dates
 - Posters
 - Summaries
 - Cast and crew information
- Users can browse and search for movies using various filters (e.g., genre, year, rating).
- The system will provide suggestions for movies based on users' search queries.

3. Watchlists and Reviews:

- Users can create and manage their own watchlists (e.g., "To Watch", "Favorites").
- Users can add movies to their watchlists from the movie database.
- Users can write and publish reviews for movies.
- Reviews will include a rating system (e.g., 1-5 stars).

4. Social Features:

- Users can follow other users and see their activity (e.g., watched movies, reviews).
- Users can like and comment on reviews.

5. Recommendations:

- The platform will provide personalized movie recommendations based on users' watchlists and ratings.
- The system will use a machine learning algorithm to generate recommendations.

UX Design Requirements:

1. User Interface:

- The platform will have a clean and simple user interface.
- The interface will be optimized for various devices (e.g., desktop, mobile, tablet).
- The platform will use a responsive design to adapt to different screen sizes.

2. Navigation:

- The platform will have a clear and consistent navigation menu.
- Users will be able to easily find and access different features (e.g., movie database, watchlists, reviews).

3. Movie Cards:

- Movie cards will display essential information about each movie (e.g., title, genre, release date, poster).
- Movie cards will be displayed in a grid or list view, depending on the user's preference.

4. Watchlist Management:

- Users will be able to easily add and remove movies from their watchlists.
- Users will be able to create and manage multiple watchlists.

5. Review Writing:

- The review writing interface will be simple and easy to use.
- Users will be able to add text, images, and ratings to their reviews.

Technical Requirements:

1. Backend:

- The platform will use a Python-based backend Flask
- The backend will be responsible for handling user input, storing data, and generating recommendations.

2. Database:

- The platform will use a relational database management system PostgreSQL
- The database will store user data, movie data, and review data.

3. Frontend:

- The platform will use a JavaScript-based frontend NextJS
- The frontend will be responsible for rendering the user interface and handling user interactions.

4. API:

- The platform will have a RESTful API for interacting with the backend.
- The API will be used to retrieve and update data.

5. Security:

- The platform will use HTTPS encryption for all data transmission.
- The platform will validate user input data to prevent SQL injection and cross-site scripting (XSS) attacks.

Infrastructure Requirements:

1. Server:

- The platform will be hosted on a cloud-based server (e.g., AWS, Google Cloud).
- The server will be configured to handle many users and data.

2. Storage:

- The platform will use a cloud-based storage solution (e.g., AWS S3, Google Cloud Storage).
- The storage solution will be used to store movie posters, user avatars, and other media files.

3. Database Server:

- The platform will use a cloud-based database server (e.g., AWS RDS, Google Cloud SQL).
- The database server will be configured to handle many queries and data.

Testing Requirements:

1. Unit Testing:

- The platform will use unit testing to ensure that individual components are working correctly.
- Unit tests will be written for the backend and frontend code.

2. Integration Testing:

- The platform will use integration testing to ensure that multiple components are working together correctly.
- Integration tests will be written for the backend and frontend code.

3. User Testing:

- The platform will use user testing to ensure that the platform is user-friendly and easy to use.
- User tests will be conducted with a small group of users to gather feedback and identify issues.

1. *Open and Closed Issues*

1 Open Issues

ID	Issue	Resolution	Responsibility	Target Date	Impact Date

2 Closed Issues

ID	Issue	Resolution	Responsibility	Target Date	Impact Date